AAAAAAAAAA....: It's almost a year now I started with fuzzing and discovered multiple bugs. The most commonly software which I've fuzzed so far includes Xpdf, VIM, PuTTY, WebKit, LibreOffice, Glibc etc. In this post I'll be demonstrating fuzzing VIM (Regex engine) through AFL++ a.k.a american fuzzy lop.

Technical Details: VIM a.k.a Vi IMproved has 12 different editing modes which can be utilized for fuzzing. Vim has lots of potential for finding bugs with AFL. One of the bug which I found while fuzzing VIM was CVE-2019-20079, I would also like to thank Dominique Pelle for this.

[+] Git clone VIM

```
cmd$ git clone https://github.com/vim/vim.git ; cd vim
```

[+] Compile and Make VIM with AFL++

```
cmd$ CC=afl-clang-fast CXX=afl-clang-fast++ ./configure --with-features=huge
--enable-gui=none
cmd$ make -j4 ; cd src/
```

[+] Feed Corpus

```
cmd$ mkdir corpus ; mkdir output
cmd$ echo "a*b\+\|[0-9]\|\d{1,9}" > corpus/1 ; echo "^\d{1,10}$" > corpus/2
```

[+] Fuzzing VIM

```
cmd$ afl-fuzz -m none -i corpus -o output ./vim -u NONE -X -Z -e -s -S @@ -c
':qa!'
```

The above options used **-u NONE** and **-X** is to speed up vim startup. Options **-e -s** are used to make vim silent and to avoid **'MORE'** prompt which could block VIM, the option **-Z** disables the external commands which makes fuzzing safer. I've also created a small bash script which automates the above tasks for you [vimfuzz.sh].

While fuzzing, fuzz it on ram file system to avoid making too much I/O something like: ***sudo mount -t tmpfs -o size=6g tmpfs /home/afl-fuzz-user/afl-fuzz***. Aside you can use [pack.sh] a script which contains some standard ubuntu packages so you dont get much dependence issues while compiling any target. Keep fuzzing :)

References:

https://www.inputzero.io/2020/03/fuzzing-vim.html
https://twitter.com/RandomDhiraj/status/1235253230625488897